PANDAS

1. Poniższy kod **uzupełnij komentarzami** odnośnie do tego jak działają poszczególne funkcje, **postaraj się zrozumieć kod.**

2. Wejdź na stronę https://www.kaggle.com/ i ściągnij plik z danymi do analizy
3. **Dokonaj analizy ściągniętych danych** ze strony https://www.kaggle.com/ za pomocą biblioteki Pandas

01_pandas_intro.ipynb

Plik Edytuj Widok Wstaw Środowisko wykonawc

Komentarz Udostępnij P

+ Kod + Tekst Połącz

```
[ ] import pandas as pd
    pd.__version__
```

```
'1.3.5'
```

W Pandas mamy dwa typy obiektów: 1. Obiekty typu Series 2. Drugi typ obiektów DataFrame - to kolekcja obiektów typu Series

I. OBIEKTY TYPU SERIES: Tworzenie obiektów cz. 1

```
[ ] s = pd.Series(data=[3, 2, 4, 6])
    s
```

```
0    3
1    2
2    4
3    6
dtype: int64
```

```
[ ] s = pd.Series(data=[3, 2, 4, 6], index=['a', 'b', 'c', 'd'] )
    s
```

```
a    3
b    2
c    4
d    6
dtype: int64
```

+ Kod + Tekst Połącz

```
[ ] s = pd.Series(data=[3, 2, 4, 6], index=['a', 'b', 'c', 'd'], name='sample')
    s
```

```
a    3
b    2
c    4
d    6
Name: sample, dtype: int64
```

```
[ ] s = pd.Series(data=[3.1, 2, 4, 6], index=['a', 'b', 'c', 'd'], name='sample')
    s
```

```
a    3.1
b    2.0
c    4.0
d    6.0
Name: sample, dtype: float64
```

Wstawianei braków danych - odwołanei do NumPy

```
[ ] import numpy as np
    s = pd.Series(data=[3.1, np.nan, 4, 6], index=['a', 'b', 'c', 'd'], name='sample'
    s
```

```
a    3.1
b    NaN
c    4.0
d    6.0
Name: sample, dtype: float64
```

Tworzenie obiektu typu bool

```
[ ] s = pd.Series(data=[True, False, False])
    s
```

```
0     True
1    False
2    False
dtype: bool
```

SEGREGACJA DANYCH po szeregach czasowych

```
[ ] s = pd.Series(data=np.arange(10, 20))
    s
```

```
0    10
1    11
2    12
3    13
4    14
5    15
6    16
7    17
8    18
9    19
dtype: int64
```

```
[ ] s = pd.Series(data=np.arange(10, 20), index=pd.date_range(start='20210322', periods=10))
    s
```

```
2021-03-22    10
2021-03-23    11
2021-03-24    12
2021-03-25    13
2021-03-26    14
2021-03-27    15
2021-03-28    16
2021-03-29    17
2021-03-30    18
2021-03-31    19
Freq: D, dtype: int64
```

OBIEKTY TYPU SERIES: Tworzenie obiektów cz. 2

```
[ ] s = pd.Series(data=['JavaScript','python', 'java'], name='languages')
    s
```

```
0    JavaScript
1        python
2          java
Name: languages, dtype: object
```

```
[ ] type(s)
```

```
pandas.core.series.Series
```

+ Kod + Tekst Połącz

```
[ ] s.index
```

```
RangeIndex(start=0, stop=3, step=1)
```

```
s = pd.Series(data=np.arange(10, 20), index=pd.date_range(start='20210322', periods=10))
s
```

```
2021-03-22    10
2021-03-23    11
2021-03-24    12
2021-03-25    13
2021-03-26    14
2021-03-27    15
2021-03-28    16
2021-03-29    17
2021-03-30    18
2021-03-31    19
Freq: D, dtype: int64
```

```
[ ] s.index
```

```
DatetimeIndex(['2021-03-22', '2021-03-23', '2021-03-24', '2021-03-25',
               '2021-03-26', '2021-03-27', '2021-03-28', '2021-03-29',
               '2021-03-30', '2021-03-31'],
              dtype='datetime64[ns]', freq='D')
```

```
[ ] list(s.index)
```

```
[Timestamp('2021-03-22 00:00:00', freq='D'),
 Timestamp('2021-03-23 00:00:00', freq='D'),
 Timestamp('2021-03-24 00:00:00', freq='D'),
 Timestamp('2021-03-25 00:00:00', freq='D'),
 Timestamp('2021-03-26 00:00:00', freq='D'),
 Timestamp('2021-03-27 00:00:00', freq='D'),
 Timestamp('2021-03-28 00:00:00', freq='D'),
 Timestamp('2021-03-29 00:00:00', freq='D'),
```

+ Kod + Tekst

```
[ ] s.values
```

```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
[ ] s.dtypes
```

```
dtype('int64')
```

```
[ ] s.shape
```

```
(10,)
```

III SERIES: PRACA Z OBIEKTAMI

```
[ ] price = pd.Series(data={'Apple': 300, 'CD Project': 70, 'Amazon': 2000}
    price
```

```
Apple          300
CD Project      70
Amazon        2000
dtype: int64
```

```
[ ] price['CD Project']
```

```
70
```

```
[ ] price[1]
```

```
70
```

```
price.count()
```

```
3
```

```
price.value_counts()
```

```
2000    1
70      1
300     1
dtype: int64
```

```
price.sum()
```

```
2370
```

```
price.min()
```

```
70
```

```
price.max()
```

```
2000
```

```
price.std()
```

```
1054.182147489122
```

```
price.describe()
```

```
count       3.000000
mean      790.000000
std      1054.182147
min        70.000000
25%       185.000000
50%       300.000000
75%      1150.000000
max      2000.000000
dtype: float64
```

```
price.describe().T
```

```
count       3.000000
mean      790.000000
std      1054.182147
min        70.000000
25%       185.000000
50%       300.000000
75%      1150.000000
max      2000.000000
dtype: float64
```

```
price = pd.Series(data={'Apple': 300, 'CD Project': 70, 'Amazon': 2000, 'KGHM': np.nan})
price
```

```
Apple          300.0
CD Project      70.0
Amazon        2000.0
KGHM             NaN
dtype: float64
```

```
price.count()
```

```
3
```

```
price.value_counts()
```

```
2000.0    1
70.0      1
300.0     1
dtype: int64
```

```
price.value_counts(dropna=False)
```

```
NaN       1
2000.0    1
70.0      1
300.0     1
dtype: int64
```

```
price.nlargest()
```

```
Amazon        2000.0
Apple          300.0
CD Project      70.0
dtype: float64
```

```
price.nlargest(2)
```

```
Amazon    2000.0
Apple      300.0
dtype: float64
```

```
price.rank()
```

```
Apple         2.0
CD Project    1.0
```

```
Amazon         3.0
KGHM           NaN
dtype: float64
```

```
price.sort_values()
```

```
CD Project      70.0
Apple          300.0
Amazon        2000.0
KGHM             NaN
dtype: float64
```

```
price.sort_values(ascending=False)
```

```
Amazon        2000.0
Apple          300.0
CD Project      70.0
KGHM             NaN
dtype: float64
```

IV. SERIES Metoda applay()

IV. SERIES Metoda applay()

```
[ ] price.apply(lambda x: x * 4.8)
```

```
[ ] price
```

```
Apple          300.0
CD Project      70.0
Amazon        2000.0
KGHM             NaN
dtype: float64
```

```
[ ] price_pln = price.apply(lambda x: x * 3.8
```

```
[ ] price_pln
```

```
Apple         1140.0
CD Project     266.0
Amazon        7600.0
KGHM             NaN
dtype: float64
```

[Bez tytułu]

```
[ ] price
```

```
Apple          300.0
CD Project      70.0
Amazon        2000.0
KGHM             NaN
dtype: float64
```

V. DATAFRAME - Tworzenie obiektów (ramki danych)

```
df = pd.DataFrame(data=[12, 12, 32])
df
```

	0
0	12
1	12
2	32

```
[ ] df = pd.DataFrame(data=[12, 12, 32], index=['first', 'second', 'third'], columns=['col_1'])
    df
```

	col_1
first	12
second	12
third	32

```
[ ] df = pd.DataFrame(data={'WIG20': ['PKN ORLEN', 'PKO SA'], 'mWIG40': ['Amica', 'Playway']})
    df
```

	WIG20	mWIG40
0	PKN ORLEN	Amica
1	PKO SA	Playway

```
[ ] df = pd.DataFrame(data=[[10, 12, 13], [23, 12, 10]], index=['first', 'second'], columns=['col_1', 'col_2', 'col_3'])
    df
```

	col_1	col_2	col_3
first	10	12	13
second	23	12	10

```
[ ] df.columns
```

```
Index(['col_1', 'col_2', 'col_3'], dtype='object')
```

```
[ ] df.index
```

```
Index(['first', 'second'], dtype='object')
```

```
[ ] df.values
```

```
array([[10, 12, 13],
       [23, 12, 10]])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2 entries, first to second
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   col_1   2 non-null      int64
 1   col_2   2 non-null      int64
 2   col_3   2 non-null      int64
dtypes: int64(3)
memory usage: 64.0+ bytes
```

```
[ ] df.describe()
```

	col_1	col_2	col_3
count	2.000000	2.0	2.00000
mean	16.500000	12.0	11.50000
std	9.192388	0.0	2.12132
min	10.000000	12.0	10.00000
25%	13.250000	12.0	10.75000
50%	16.500000	12.0	11.50000
75%	19.750000	12.0	12.25000
max	23.000000	12.0	13.00000

```
[ ] df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
col_1	2.0	16.5	9.192388	10.0	13.25	16.5	19.75	23.0
col_2	2.0	12.0	0.000000	12.0	12.00	12.0	12.00	12.0
col_3	2.0	11.5	2.121320	10.0	10.75	11.5	12.25	13.0

VI. DATAFRAME SELEKCJA - WYCINANIE KOLUMN

```
df
```

	col_1	col_2	col_3
first	10	12	13
second	23	12	10

```
df['col_1']
```

```
first     10
second    23
Name: col_1, dtype: int64
```

```
type(df['col_1'])
```

```
pandas.core.series.Series
```

```
df[['col_1']]
```

	col_1
first	10
second	23

```
type(df[['col_1']])
```

```
pandas.core.frame.DataFrame
```

```
df.columns
```

```
Index(['col_1', 'col_2', 'col_3'], dtype='object')
```

```
df.columns = ['a', 'b', 'c']
df
```

	a	b	c
first	10	12	13
second	23	12	10

```
df.a
```

```
first     10
second    23
Name: a, dtype: int64
```

```
df.b
```

```
first     12
second    12
Name: b, dtype: int64
```

```
df.columns = ['a', 'pawel n', 'c']
df
```

	a	pawel n	c
first	10	12	13
second	23	12	10

```
df.columns = ['a', 'pawel_n', 'c']
df
```

	a	pawel_n	c
first	10	12	13
second	23	12	10

```
df.pawel_n
```

```
first     12
second    12
Name: pawel_n, dtype: int64
```

```
df['d'] = df.a + df.c
df
```

	a	pawel_n	c	d
first	10	12	13	23
second	23	12	10	33

```python
df = pd.DataFrame(data=[[10, 12, 13], [23, 12, 10]], index=['first', 'second'], columns=['col_1', 'col_2', 'col_3'])
df
```

	col_1	col_2	col_3
first	10	12	13
second	23	12	10

```python
df.loc['first']
```

```
col_1    10
col_2    12
col_3    13
Name: first, dtype: int64
```

```python
df.iloc[0]
```

```
col_1    10
col_2    12
col_3    13
Name: first, dtype: int64
```

```python
df.loc['first', 'col_2']
```

```
12
```

```python
df.loc[:, 'col_2']
```

```
first     12
second    12
Name: col_2, dtype: int64
```

```python
df.iloc[0, 1]
```

```
12
```

VI. CASE STUDY - WYKORZYSTANIE UMIEJETNOSCI W PRAKTYCE

1. IMPORT BIBLIOTEK

```
[ ] import numpy as np
    import pandas as pd

    print(f'Numpy: {np.__version__}')
    print(f'Pandas: {pd.__version__}')
```

```
Numpy: 1.19.5
Pandas: 1.1.5
```

2. Załadowanie danych za pomoca funkcji fetch...

```
[ ] def fetch_financial_data(company='AMZN'):
      """Pobierane dane dotyczą notowań spółek na giełdzie wg symboli na serwisie stooq.pl"""
      import pandas_datareader.data as web
      return web.DataReader(name=company, data_source='stooq')

    df = fetch_financial_data()
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1258 entries, 2021-03-24 to 2016-03-28
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Open    1258 non-null   float64
 1   High    1258 non-null   float64
 2   Low     1258 non-null   float64
 3   Close   1258 non-null   float64
 4   Volume  1258 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 59.0 KB
```

```
df
```

	Open	High	Low	Close	Volume
Date					
2021-03-24	3151.04	3160.310	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.000	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.580	3060.0500	3110.87	2902179
2021-03-19	3029.23	3077.285	3016.6300	3074.96	4625354
2021-03-18	3101.00	3116.630	3025.0000	3027.99	3657027
...	...	...	...	...	...
2016-04-01	590.49	599.030	588.3000	598.50	2293292
2016-03-31	599.28	600.750	592.2100	593.64	2032513
2016-03-30	596.71	603.240	595.0000	598.69	3204413
2016-03-29	580.15	595.850	576.5000	593.86	3702016
2016-03-28	584.40	584.750	575.5600	579.87	2711908

1258 rows × 5 columns

Dodaj komórkę z tekstem

```
[ ] df.head()
```

	Open	High	Low	Close	Volume
Date					
2021-03-24	3151.04	3160.310	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.000	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.580	3060.0500	3110.87	2902179
2021-03-19	3029.23	3077.285	3016.6300	3074.96	4625354
2021-03-18	3101.00	3116.630	3025.0000	3027.99	3657027

+ Kod + Tekst

```
df.head(3)
```

	Open	High	Low	Close	Volume
Date					
2021-03-24	3151.04	3160.31	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.00	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.58	3060.0500	3110.87	2902179

```
df.tail()
```

	Open	High	Low	Close	[Bez tytułu]
Date					
2016-04-01	590.49	599.03	588.30	598.50	2293292
2016-03-31	599.28	600.75	592.21	593.64	2032513
2016-03-30	596.71	603.24	595.00	598.69	3204413
2016-03-29	580.15	595.85	576.50	593.86	3702016
2016-03-28	584.40	584.75	575.56	579.87	2711908

```
df.tail(3)
```

	Open	High	Low	Close	Volume
Date					
2016-03-30	596.71	603.24	595.00	598.69	3204413
2016-03-29	580.15	595.85	576.50	593.86	3702016
2016-03-28	584.40	584.75	575.56	579.87	2711908

+ Kod + Te

```
df.columns
```

```
Index(['Open', 'High', 'Low', 'Close', 'Volume'], dtype='object')
```

```
df.columns = [col.lower() for col in df.columns]
df.head()
```

	open	high	low	close	volume
Date					
2021-03-24	3151.04	3160.310	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.000	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.580	3060.0500	3110.87	2902179
2021-03-19	3029.23	3077.285	3016.6300	3074.96	4625354
2021-03-18	3101.00	3116.630	3025.0000	3027.99	3657027

```
df.describe()
```

	open	high	low	close	volume
count	1258.000000	1258.000000	1258.000000	1258.000000	1.258000e+03
mean	1675.007771	1692.427869	1655.088469	1674.325131	4.239223e+06
std	786.242914	796.959564	773.484289	784.861964	2.117987e+06
min	580.150000	584.750000	575.560000	579.870000	8.813370e+05
25%	967.330000	974.422500	959.195000	965.942500	2.817824e+06
50%	1673.995000	1697.595000	1652.080000	1670.595000	3.673067e+06
75%	1930.670000	1950.740000	1906.870000	1926.495000	5.095006e+06
max	3547.000000	3552.250000	3486.685000	3531.450000	1.655260e+07

+ Kod + Tekst

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
open	1258.0	1.675008e+03	7.862429e+02	580.15	9.673300e+02	1673.995	1930.670	3.547000e+03
high	1258.0	1.692428e+03	7.969596e+02	584.75	9.744225e+02	1697.595	1950.740	3.552250e+03
low	1258.0	1.655088e+03	7.734843e+02	575.56	9.591950e+02	1652.080	1906.870	3.486685e+03
close	1258.0	1.674325e+03	7.848620e+02	579.87	9.659425e+02	1670.595	1926.495	3.531450e+03
volume	1258.0	4.239223e+06	2.117987e+06	881337.00	2.817824e+06	3673067.000	5095005.750	1.655260e+07

3. SELEKCJA KOLUMN - jak wycinać kolumny

```
df = df.head(10)
df
```

	open	high	low	close	volume
Date					
2021-03-24	3151.04	3160.3100	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.0000	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.5800	3060.0500	3110.87	2902179
2021-03-19	3029.23	3077.2850	3016.6300	3074.96	4625354
2021-03-18	3101.00	3116.6300	3025.0000	3027.99	3657027
2021-03-17	3073.22	3173.0500	3070.2200	3135.73	3118584
2021-03-16	3104.97	3128.9100	3075.8601	3091.86	2538764
2021-03-15	3074.57	3082.2400	3032.0900	3081.68	2918592
2021-03-12	3075.00	3098.9800	3045.5000	3089.49	2421888
2021-03-11	3104.01	3131.7843	3082.9300	3113.59	2776391

```
df['open']
```

```
Date
2021-03-24    3151.04
2021-03-23    3127.00
2021-03-22    3067.85
2021-03-19    3029.23
2021-03-18    3101.00
2021-03-17    3073.22
2021-03-16    3104.97
2021-03-15    3074.57
2021-03-12    3075.00
2021-03-11    3104.01
Name: open, dtype: float64
```

```
df.open
```

```
Date
2021-03-24    3151.04
2021-03-23    3127.00
2021-03-22    3067.85
2021-03-19    3029.23
2021-03-18    3101.00
2021-03-17    3073.22
2021-03-16    3104.97
2021-03-15    3074.57
2021-03-12    3075.00
2021-03-11    3104.01
Name: open, dtype: float64
```

```
df[['open']]
```

	open
Date	
2021-03-24	3151.04
2021-03-23	3127.00
2021-03-22	3067.85
2021-03-19	3029.23
2021-03-18	3101.00
2021-03-17	3073.22
2021-03-16	3104.97
2021-03-15	3074.57
2021-03-12	3075.00
2021-03-11	3104.01

```
df[['open', 'close']]
```

	open	close
Date		
2021-03-24	3151.04	3087.07
2021-03-23	3127.00	3137.50
2021-03-22	3067.85	3110.87
2021-03-19	3029.23	3074.96
2021-03-18	3101.00	3027.99
2021-03-17	3073.22	3135.73
2021-03-16	3104.97	3091.86
2021-03-15	3074.57	3081.68
2021-03-12	3075.00	3089.49
2021-03-11	3104.01	3113.59

```
df.iloc[:, 0]
```

```
Date
2021-03-24    3151.04
2021-03-23    3127.00
2021-03-22    3067.85
2021-03-19    3029.23
2021-03-18    3101.00
2021-03-17    3073.22
2021-03-16    3104.97
2021-03-15    3074.57
2021-03-12    3075.00
2021-03-11    3104.01
Name: open, dtype: float64
```

```
df.iloc[:, [0, 3]]
```

	open	close
Date		
2021-03-24	3151.04	3087.07
2021-03-23	3127.00	3137.50
2021-03-22	3067.85	3110.87
2021-03-19	3029.23	3074.96
2021-03-18	3101.00	3027.99
2021-03-17	3073.22	3135.73
2021-03-16	3104.97	3091.86
2021-03-15	3074.57	3081.68
2021-03-12	3075.00	3089.49
2021-03-11	3104.01	3113.59

```
df.iloc[:, 3:5]
```

	close	volume
Date		
2021-03-24	3087.07	2952085
2021-03-23	3137.50	3817263
2021-03-22	3110.87	2902179
2021-03-19	3074.96	4625354
2021-03-18	3027.99	3657027
2021-03-17	3135.73	3118584
2021-03-16	3091.86	2538764
2021-03-15	3081.68	2918592
2021-03-12	3089.49	2421888
2021-03-11	3113.59	2776391

```
df.iloc[:, -1]
```

```
Date
2021-03-24    2952085
2021-03-23    3817263
2021-03-22    2902179
2021-03-19    4625354
2021-03-18    3657027
2021-03-17    3118584
2021-03-16    2538764
2021-03-15    2918592
2021-03-12    2421888
2021-03-11    2776391
Name: volume, dtype: int64
```

```
[ ] df.iloc[:, [-1]]
```

Date	volume
2021-03-24	2952085
2021-03-23	3817263
2021-03-22	2902179
2021-03-19	4625354
2021-03-18	3657027
2021-03-17	3118584
2021-03-16	2538764
2021-03-15	2918592
2021-03-12	2421888
2021-03-11	2776391

```
df.iloc[:, -3:]
```

Date	low	close	volume
2021-03-24	3085.1500	3087.07	2952085
2021-03-23	3120.8501	3137.50	3817263
2021-03-22	3060.0500	3110.87	2902179
2021-03-19	3016.6300	3074.96	4625354
2021-03-18	3025.0000	3027.99	3657027
2021-03-17	3070.2200	3135.73	3118584
2021-03-16	3075.8601	3091.86	2538764
2021-03-15	3032.0900	3081.68	2918592
2021-03-12	3045.5000	3089.49	2421888
2021-03-11	3082.9300	3113.59	2776391

4. SELEKCJA WIERSZY

```
[ ] df.iloc[0]
```

```
open         3151.04
high         3160.31
low          3085.15
close        3087.07
volume    2952085.00
Name: 2021-03-24 00:00:00, dtype: float64
```

```
[ ] df.iloc[:3]
```

Date	open	high	low	close	volume
2021-03-24	3151.04	3160.31	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.00	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.58	3060.0500	3110.87	2902179

```
[ ] df
```

	open	high	low	close	volume
Date					
2021-03-24	3151.04	3160.3100	3085.1500	3087.07	2952085
2021-03-23	3127.00	3182.0000	3120.8501	3137.50	3817263
2021-03-22	3067.85	3126.5800	3060.0500	3110.87	2902179
2021-03-19	3029.23	3077.2850	3016.6300	3074.96	4625354
2021-03-18	3101.00	3116.6300	3025.0000	3027.99	3657027
2021-03-17	3073.22	3173.0500	3070.2200	3135.73	3118584
2021-03-16	3104.97	3128.9100	3075.8601	3091.86	2538764
2021-03-15	3074.57	3082.2400	3032.0900	3081.68	2918592
2021-03-12	3075.00	3098.9800	3045.5000	3089.49	2421888
2021-03-11	3104.01	3131.7843	3082.9300	3113.59	2776391

```
[ ] df.loc['2021-03-22':'2021-03-18']
```

	open	high	low	close	volume
Date					
2021-03-22	3067.85	3126.580	3060.05	3110.87	2902179
2021-03-19	3029.23	3077.285	3016.63	3074.96	4625354
2021-03-18	3101.00	3116.630	3025.00	3027.99	3657027

7. DATAFRAME - OBLICZANIE NOWYCH KOLUMN

```
[ ] df = fetch_financial_data('MSFT')
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1258 entries, 2021-03-24 to 2016-03-28
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Open    1258 non-null   float64
 1   High    1258 non-null   float64
 2   Low     1258 non-null   float64
 3   Close   1258 non-null   float64
 4   Volume  1258 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 59.0 KB
```

```
[ ] df = fetch_financial_data('UBER')
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 472 entries, 2021-03-24 to 2019-05-10
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Open    472 non-null    float64
 1   High    472 non-null    float64
 2   Low     472 non-null    float64
 3   Close   472 non-null    float64
 4   Volume  472 non-null    int64
dtypes: float64(4), int64(1)
memory usage: 22.1 KB
```

+ Kod + Tekst

```
[ ] df.head()
```

	Open	High	Low	Close	Volume
Date					
2021-03-24	54.46	54.96	52.54	52.57	13689068
2021-03-23	55.63	55.63	53.11	53.49	14513457
2021-03-22	57.25	57.61	55.47	55.69	14149412
2021-03-19	55.48	57.18	54.34	57.08	17396297
2021-03-18	56.63	57.48	55.21	55.69	15742985

```
[ ] df.describe()
```

	Open	High	Low	Close	Volume
count	472.000000	472.000000	472.000000	472.000000	4.720000e+02
mean	37.985050	38.753932	37.100169	37.947352	2.400973e+07
std	9.508638	9.607251	9.342303	9.450463	1.827766e+07
min	15.960000	17.800000	13.710000	14.820000	3.380003e+06
25%	31.167500	31.815000	30.480000	31.165000	1.311506e+07
50%	34.605000	35.199650	33.960000	34.585000	1.959129e+07
75%	43.620000	44.252500	43.035000	43.692500	2.959548e+07
max	63.250000	64.050000	60.800000	63.180000	1.863225e+08

```
df.head(3)
```

	Open	High	Low	Close	Volume
Date					
2021-03-24	54.46	54.96	52.54	52.57	13689068
2021-03-23	55.63	55.63	53.11	53.49	14513457
2021-03-22	57.25	57.61	55.47	55.69	14149412

```
[ ] df['Average'] = (df.Open + df.Close) /2.0
    df.head()
```

	Open	High	Low	Close	Volume	Average
Date						
2021-03-24	54.46	54.96	52.54	52.57	13689068	53.515
2021-03-23	55.63	55.63	53.11	53.49	14513457	54.560
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.470
2021-03-19	55.48	57.18	54.34	57.08	17396297	56.280
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.160

```
df.sort_index()
```

	Open	High	Low	Close	Volume	Average
Date						
2019-05-10	42.00	45.00	41.06	41.57	186322536	41.785
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240
...	...	...	...	...	...	...
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.160
2021-03-19	55.48	57.18	54.34	57.08	17396297	56.280
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.470
2021-03-23	55.63	55.63	53.11	53.49	14513457	54.560
2021-03-24	54.46	54.96	52.54	52.57	13689068	53.515

472 rows × 6 columns

```
df = df.sort_index()
df.head()
```

	Open	High	Low	Close	Volume	Average
Date						
2019-05-10	42.00	45.00	41.06	41.57	186322536	41.785
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240

```
df[['Close']].shift(1)
```

	Close
Date	
2019-05-10	NaN
2019-05-13	41.57
2019-05-14	37.10
2019-05-15	39.96
2019-05-16	41.29
...	...
2021-03-18	56.36
2021-03-19	55.69
2021-03-22	57.08
2021-03-23	55.69
2021-03-24	53.49

472 rows × 1 columns

```
df[['Close']].shift(3)
```

	Close
Date	
2019-05-10	NaN
2019-05-13	NaN
2019-05-14	NaN
2019-05-15	41.57
2019-05-16	37.10
...	...
2021-03-18	60.19
2021-03-19	58.85
2021-03-22	56.36
2021-03-23	55.69
2021-03-24	57.08

```
df['Close_shift'] = df.Close.shift(1)
df.head()
```

	Open	High	Low	Close	Volume	Average	Close_shift
Date							
2019-05-10	42.00	45.00	41.06	41.57	186322536	41.785	NaN
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945	41.57
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135	37.10
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330	39.96
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240	41.29

```
df. Close / df.Close_shift -1
```

```
Date
2019-05-10         NaN
2019-05-13   -0.107529
2019-05-14    0.077089
2019-05-15    0.033283
2019-05-16    0.041414
                 ...
2021-03-18   -0.011888
2021-03-19    0.024960
2021-03-22   -0.024352
2021-03-23   -0.039504
2021-03-24   -0.017199
Length: 472, dtype: float64
```

```
[ ] df['Daily_Change'] = df. Close / df.Close_shift -1
    df.head()
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2019-05-10	42.00	45.00	41.06	41.57	186322536	41.785	NaN	NaN
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945	41.57	-0.107529
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135	37.10	0.077089
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330	39.96	0.033283
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240	41.29	0.041414

Aby edytować zawartość komórki, kliknij ją dwukrotnie (lub naciśnij klawisz Enter)

del df['Daily Change'] df.head() WRZUĆ W TYM MIEJSCU DO KODU

```
[ ] df.Daily_Change.min()
```

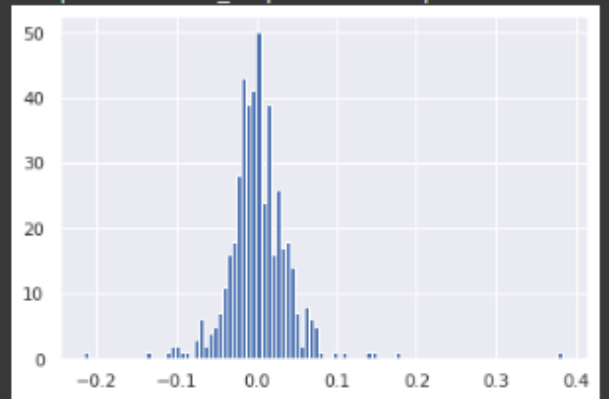
```
-0.21628767847699626
```

```
[ ] df.Daily_Change.max()
```

```
0.3825910931174088
```

```
[ ] df.Daily_Change.hist(bins=100)
```

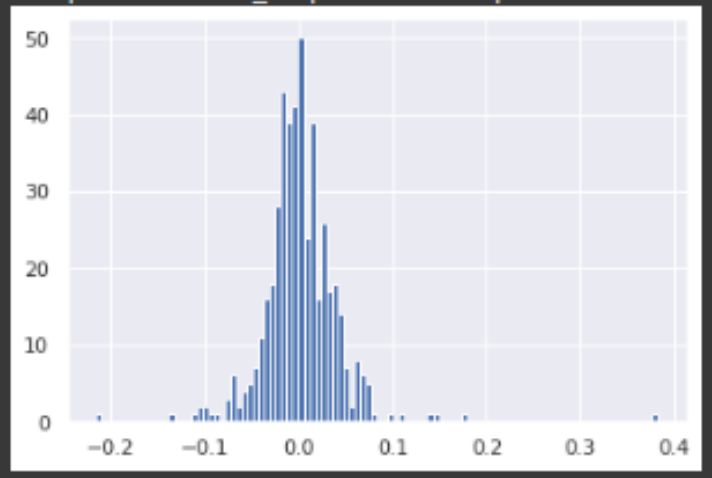
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe18b7538d0>
```




```
import seaborn as sns
sns.set()

df.Daily_Change.hist(bins=100)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe18b508d90>
```




```
[ ] df.Close.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe18b3dd750>
```

VIII. DataFrame - FILTROWANIE DANYCH

```
df
```

Date	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
2019-05-10	42.00	45.00	41.06	41.57	186322536	41.785	NaN	NaN
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945	41.57	-0.107529
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135	37.10	0.077089
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330	39.96	0.033283
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240	41.29	0.041414
...	...	...	...	...	...	...	...	...
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.160	56.36	-0.011888
2021-03-19	55.48	57.18	54.34	57.08	17396297	56.280	55.69	0.024960
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.470	57.08	-0.024352
2021-03-23	55.63	55.63	53.11	53.49	14513457	54.560	55.69	-0.039504
2021-03-24	54.46	54.96	52.54	52.57	13689068	53.515	53.49	-0.017199

472 rows × 8 columns

```
[ ] df.Daily_Change > 0
```

```
Date
2019-05-10    False
2019-05-13    False
2019-05-14     True
2019-05-15     True
2019-05-16     True
              ...
2021-03-18    False
2021-03-19     True
2021-03-22    False
2021-03-23    False
2021-03-24    False
Name: Daily_Change, Length: 472, dtype: bool
```

```
df[df.Daily_Change > 0]
```

Date	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135	37.10	0.077089
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330	39.96	0.033283
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240	41.29	0.041414
2019-05-24	41.28	41.51	40.50	41.51	8786751	41.395	40.47	0.025698
2019-05-31	41.15	41.57	39.41	40.41	23209848	40.780	39.80	0.015327
...	...	...	...	...	...	...	...	...
2021-03-09	56.00	56.15	54.72	55.25	17192937	55.625	53.20	0.038534
2021-03-10	57.21	58.90	56.27	57.68	27997015	57.445	55.25	0.043982
2021-03-11	58.80	59.48	58.22	58.95	17293752	58.875	57.68	0.022018
2021-03-12	58.97	60.59	57.62	60.35	14839307	59.660	58.95	0.023749
2021-03-19	55.48	57.18	54.34	57.08	17396297	56.280	55.69	0.024960

228 rows × 8 columns

```
[ ] df[df.Daily_Change < 0]
```

Date	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945	41.57	-0.107529
2019-05-17	41.98	43.29	41.27	41.91	20225687	41.945	43.00	-0.025349
2019-05-20	41.19	41.68	39.46	41.59	29222330	41.390	41.91	-0.007635
2019-05-21	42.00	42.24	41.25	41.50	10802851	41.750	41.59	-0.002164
2019-05-22	41.05	41.28	40.50	41.25	9089469	41.150	41.50	-0.006024
...	...	...	...	...	...	...	...	...
2021-03-17	57.07	57.57	55.54	56.36	25509926	56.715	58.85	-0.042311
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.160	56.36	-0.011888
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.470	57.08	-0.024352
2021-03-23	55.63	55.63	53.11	53.49	14513457	54.560	55.69	-0.039504

```
df_positive = df[df.Daily_Change > 0]
df_positive
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135	37.10	0.077089
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330	39.96	0.033283
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240	41.29	0.041414
2019-05-24	41.28	41.51	40.50	41.51	8786751	41.395	40.47	0.025698
2019-05-31	41.15	41.57	39.41	40.41	23209848	40.780	39.80	0.015327
...	...	...	...	...	...	...	...	...
2021-03-09	56.00	56.15	54.72	55.25	17192937	55.625	53.20	0.038534
2021-03-10	57.21	58.90	56.27	57.68	27997015	57.445	55.25	0.043982
2021-03-11	58.80	59.48	58.22	58.95	17293752	58.875	57.68	0.022018
2021-03-12	58.97	60.59	57.62	60.35	14839307	59.660	58.95	0.023749
2021-03-19	55.48	57.18	54.34	57.08	17396297	56.280	55.69	0.024960

228 rows × 8 columns

```
df_positive.Daily_Change.mean()
```

```
0.028679292384398338
```

```
df_negative = df[df.Daily_Change < 0]
df_negative
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945	41.57	-0.107529
2019-05-17	41.98	43.29	41.27	41.91	20225687	41.945	43.00	-0.025349
2019-05-20	41.19	41.68	39.46	41.59	29222330	41.390	41.91	-0.007635
2019-05-21	42.00	42.24	41.25	41.50	10802851	41.750	41.59	-0.002164
2019-05-22	41.05	41.28	40.50	41.25	9089469	41.150	41.50	-0.006024
...	...	...	...	...	...	...	...	...
2021-03-17	57.07	57.57	55.54	56.36	25509926	56.715	58.85	-0.042311
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.160	56.36	-0.011888
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.470	57.08	-0.024352

```
df_negative.Daily_Change.mean()
```

```
-0.024710733720425187
```

```
df.Close == df.High
```

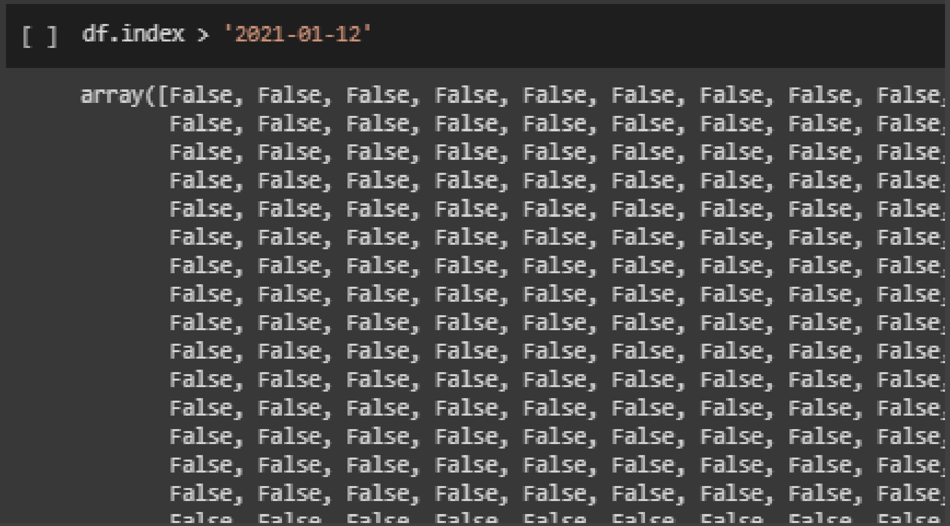
```
Date
2019-05-10    False
2019-05-13    False
2019-05-14     True
2019-05-15    False
2019-05-16    False
              ...
2021-03-18    False
2021-03-19    False
2021-03-22    False
2021-03-23    False
2021-03-24    False
Length: 472, dtype: bool
```

```
df[df.Close == df.Low]
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2019-08-12	39.84	39.95	37.00	37.00	20632310	38.420	40.05	-0.076155
2019-12-09	27.96	28.36	27.68	27.68	21098387	27.820	27.86	-0.006461
2019-12-27	30.80	31.06	30.17	30.17	18465965	30.485	30.67	-0.016303
2020-12-31	53.28	53.28	51.00	51.00	13282786	52.140	53.15	-0.040452

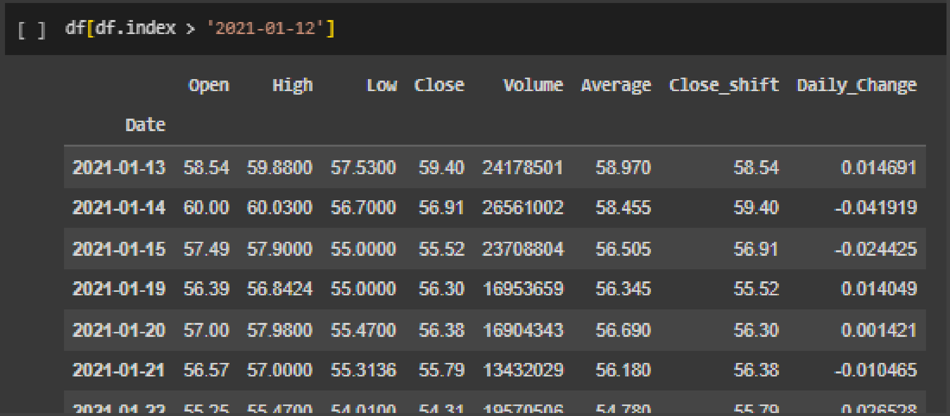
```
df.index
```

```
DatetimeIndex(['2019-05-10', '2019-05-13', '2019-05-14', '2019-05-15',
               '2019-05-16', '2019-05-17', '2019-05-20', '2019-05-21',
               '2019-05-22', '2019-05-23',
               ...
               '2021-03-11', '2021-03-12', '2021-03-15', '2021-03-16',
               '2021-03-17', '2021-03-18', '2021-03-19', '2021-03-22',
               '2021-03-23', '2021-03-24'],
              dtype='datetime64[ns]', name='Date', length=472, freq=None)
```



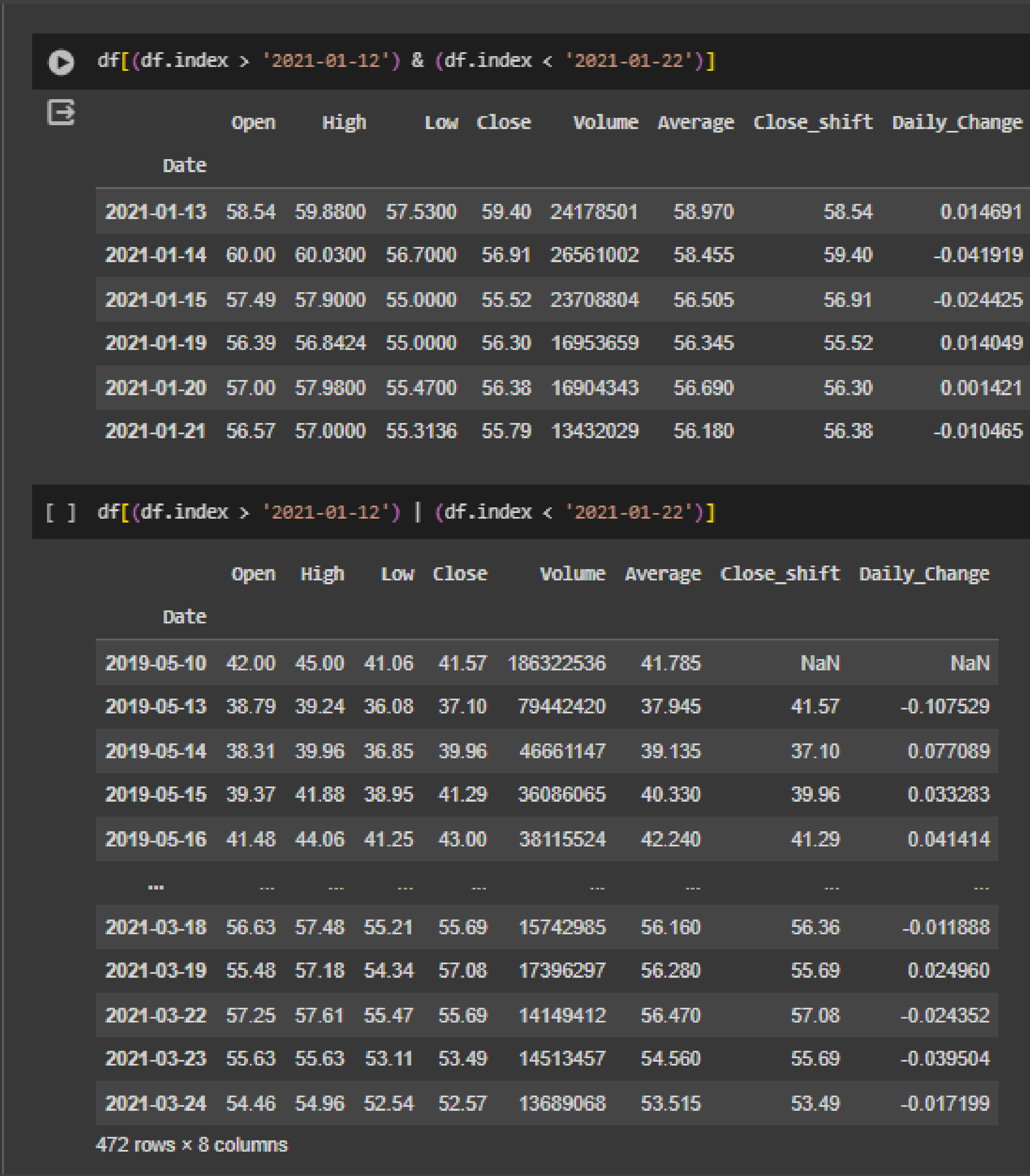
```
[ ] df.index > '2021-01-12'

    array([False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False,
           False, False, False, False, False, False, False, False, False
```



```
[ ] df[df.index > '2021-01-12']

               Open     High      Low  Close    Volume  Average  Close_shift  Daily_Change
         Date
   2021-01-13  58.54  59.8800  57.5300  59.40  24178501   58.970        58.54      0.014691
   2021-01-14  60.00  60.0300  56.7000  56.91  26561002   58.455        59.40     -0.041919
   2021-01-15  57.49  57.9000  55.0000  55.52  23708804   56.505        56.91     -0.024425
   2021-01-19  56.39  56.8424  55.0000  56.30  16953659   56.345        55.52      0.014049
   2021-01-20  57.00  57.9800  55.4700  56.38  16904343   56.690        56.30      0.001421
   2021-01-21  56.57  57.0000  55.3136  55.79  13432029   56.180        56.38     -0.010465
   2021-01-22  55.25  55.4700  54.0100  54.31  19570506   54.780        55.79      0.026528
```



```
df[(df.index > '2021-01-12') & (df.index < '2021-01-22')]

               Open     High      Low  Close    Volume  Average  Close_shift  Daily_Change
         Date
   2021-01-13  58.54  59.8800  57.5300  59.40  24178501   58.970        58.54      0.014691
   2021-01-14  60.00  60.0300  56.7000  56.91  26561002   58.455        59.40     -0.041919
   2021-01-15  57.49  57.9000  55.0000  55.52  23708804   56.505        56.91     -0.024425
   2021-01-19  56.39  56.8424  55.0000  56.30  16953659   56.345        55.52      0.014049
   2021-01-20  57.00  57.9800  55.4700  56.38  16904343   56.690        56.30      0.001421
   2021-01-21  56.57  57.0000  55.3136  55.79  13432029   56.180        56.38     -0.010465
```

```
[ ] df[(df.index > '2021-01-12') | (df.index < '2021-01-22')]

               Open   High    Low  Close     Volume  Average  Close_shift  Daily_Change
         Date
   2019-05-10  42.00  45.00  41.06  41.57  186322536   41.785          NaN           NaN
   2019-05-13  38.79  39.24  36.08  37.10   79442420   37.945        41.57     -0.107529
   2019-05-14  38.31  39.96  36.85  39.96   46661147   39.135        37.10      0.077089
   2019-05-15  39.37  41.88  38.95  41.29   36086065   40.330        39.96      0.033283
   2019-05-16  41.48  44.06  41.25  43.00   38115524   42.240        41.29      0.041414
          ...    ...    ...    ...    ...        ...      ...          ...           ...
   2021-03-18  56.63  57.48  55.21  55.69   15742985   56.160        56.36     -0.011888
   2021-03-19  55.48  57.18  54.34  57.08   17396297   56.280        55.69      0.024960
   2021-03-22  57.25  57.61  55.47  55.69   14149412   56.470        57.08     -0.024352
   2021-03-23  55.63  55.63  53.11  53.49   14513457   54.560        55.69     -0.039504
   2021-03-24  54.46  54.96  52.54  52.57   13689068   53.515        53.49     -0.017199

  472 rows × 8 columns
```

```
[ ] df[(df.index == '2021-03-22') | (df.index == '2021-03-18')]
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.16	56.36	-0.011888
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.47	57.08	-0.024352

```
[ ] df
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2019-05-10	42.00	45.00	41.06	41.57	186322536	41.785	NaN	NaN
2019-05-13	38.79	39.24	36.08	37.10	79442420	37.945	41.57	-0.107529
2019-05-14	38.31	39.96	36.85	39.96	46661147	39.135	37.10	0.077089
2019-05-15	39.37	41.88	38.95	41.29	36086065	40.330	39.96	0.033283
2019-05-16	41.48	44.06	41.25	43.00	38115524	42.240	41.29	0.041414
...	...	...	...	...	...	...	...	...
2021-03-18	56.63	57.48	55.21	55.69	15742985	56.160	56.36	-0.011888
2021-03-19	55.48	57.18	54.34	57.08	17396297	56.280	55.69	0.024960
2021-03-22	57.25	57.61	55.47	55.69	14149412	56.470	57.08	-0.024352
2021-03-23	55.63	55.63	53.11	53.49	14513457	54.560	55.69	-0.039504
2021-03-24	54.46	54.96	52.54	52.57	13689068	53.515	53.49	-0.017199

472 rows × 8 columns

```
[ ] df.index.month
```

```
Int64Index([5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
            ...
            3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
           dtype='int64', name='Date', length=472)
```

```
df.index.month == 5
```

```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
```

```
[ ] df[df.index.year == 2021]
```

	Open	High	Low	Close	Volume	Average	Close_shift	Daily_Change
Date								
2021-01-04	52.220	52.3200	49.6350	51.14	17291804	51.6800	51.00	0.002745
2021-01-05	51.000	54.1900	50.7600	54.01	21403169	52.5050	51.14	0.056120
2021-01-06	53.310	54.0700	52.0000	52.48	17738115	52.8950	54.01	-0.028328
2021-01-07	53.370	56.2500	53.2000	56.13	23737543	54.7500	52.48	0.069550
2021-01-08	54.395	54.7900	52.5800	53.28	37999442	53.8375	56.13	-0.050775
2021-01-11	53.110	55.0000	52.9900	54.59	23213840	53.8500	53.28	0.024587
2021-01-12	55.500	59.3900	55.0000	58.54	52151405	57.0200	54.59	0.072358